Visualization of Boundary Solutions of A High Dimensional Pareto-front from A Decision Maker's Perspective

Author1
REDACTED Laboratory
REDACTED University
CITY, *STATE*, *COUNTRY*
email1

Author2
REDACTED Laboratory
REDACTED University
CITY, *STATE*, *COUNTRY*
email2

ABSTRACT

For the last couple of years, the development of many-objective optimization problems opened new avenues of research in the evolutionary multi-objective optimization domain. There are already a number of algorithms to solve such problems, now the next challenge is to interpret the results produced by those algorithms. In this paper, we propose an alternative way to visualize high dimensional Pareto-front where the goal is to present the Pareto-front in terms of a decision maker's perspective. All the existing Pareto-from visualization approaches emphasize on the algorithm convergence speed and quality (i.e. convergence and spread) of the Pareto solutions. However, such information is rarely useful in a typical decision making phase. A decision maker is more interested in the different aspects of the end result. They are interested in Pareto-optimal solutions that offer the most trade-off. They are also interested to know the relative robustness of a solution and how their neighborhood looks like. In this paper, we present a way to visualize the Pareto-front in high dimension by keeping those criteria in mind. Our approach can present the high dimensional Pareto-front in a way similar to that of a scatter plot. Thus facilitating the decision making process more human centric.

CCS CONCEPTS

• **Applied computing** → **Multi-criterion optimization and decision-making**; • **Human-centered computing** → *Visualization theory, concepts and paradigms*;

KEYWORDS

Many-objective Optimization, Pareto-front, Visualization, Decision Making, Neighborhood Embedding, Knee Points, Boundary Solutions, Cluster

ACM Reference Format:
Author1 and *Author2*. 2018. Visualization of Boundary Solutions of A High Dimensional Pareto-front from A Decision Maker's Perspective. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018 (GECCO '18).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the recent advancement in many-objective (i.e. problems with four or more conflicting objective functions) optimization algorithms, there have been a number of post-optimization issues that need to be addressed. One of the most challenging problem is to understand the results produced by a many-objective optimization (MOOP) solver. In an MOOP scenario, if the target problem is consisted of two or three objectives, a scatter plot is the most intuitive way to visualize the results, which allows the *decision maker (DM)* to easily understand the trade-off between the objectives, robustness of a solution (i.e. risk assessment) or analyzing the neighborhood of a particular solutions etc. Naturally, since it is not possible for a DM to comprehend four or more spatial dimensions visually, we try to interpret data points from higher dimensions by mapping them over a lower dimensional space. One of the most commonly used such techniques is the Parallel Coordinate Plot (PCP) [8], another such example is RadViz [7]. However, all such methods do not necessarily satisfy all the criteria required for a better decision making in an MOOP setting, since those methods were developed for keeping other criteria in mind. The goal of this paper is to discuss some alternatives that allows a decision maker to easily explore the results of an MOOP. This is a very important topic, since without a straight forward and intuitive interpretation of the results, all the attempts to solve a difficult MOOP become somewhat useless, specially in terms of real world engineering applications.

In terms of high-dimensional and multivariate data analysis, there have been a number of different visualization methods invented. Some of which have been applied to the visualization of MOOP solution sets. As we have already mentioned, one of the most widely used methods is the PCP [8], however this technique suffers from several limitations. The most difficult one is the optimal arrangement of the parallel vertical axes. Without the optimal arrangement of those axes, a meaningful comprehension of the results becomes difficult. Another approach is so called Heatmaps, which presents a visualization from which all of the original

GECCO '18, July 15–19, 2018, Kyoto, Japan
© 2018 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

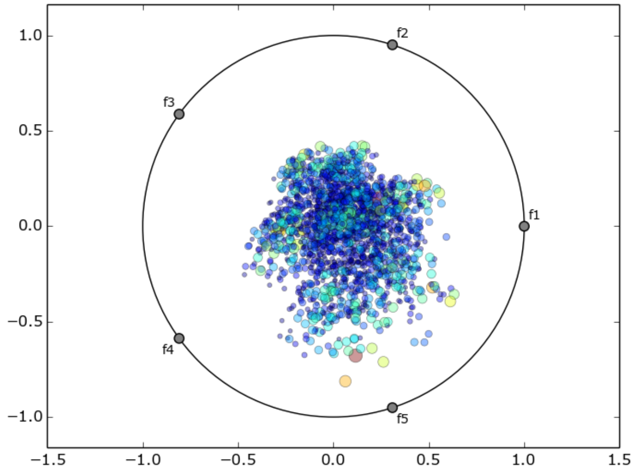


Figure 1: A RadViz plot of a 5 objective DTLZ6 problem, where the true Pareto front (PF) is consisted of five different clusters, however RadViz superimpose all clusters into one place hence there is no way to visualize them separately.

data can be recovered programmatically; however, they are generally difficult to interpret because solutions are overlaid or arbitrarily ordered. Other basic transformation methods, such as principal component analysis (PCA), transforms the dimensionality of the data points into a 2-D space so that it can be visualized with standard techniques, such as scatter plots. The main problem of these transformation based technique is their limitations to recover the original objective values that have been used as inputs.

In RadViz [7], each dimension in the dataset is represented by a dimensional anchor, and each dimensional anchor is distributed evenly on a unit circle. Each line in the data set corresponds to a point in the projection, that is linked to every dimensional anchor by a spring. Each spring's stiffness corresponds to the value for that particular thing in that particular dimension. The position of the point is defined as the point in the 2D space where the spring's tension is minimum. that transforms the objective vectors to map over polar coordinate in a specific way. However, RadViz fails to illustrate the Pareto-front (PF) completely when there are multiple cluster of solutions in the high dimensional space. One such example is presented in Figure 1.

Despite the loss of important information relevant to decision making, dimension-reduction methods often offer a useful visualization and it is especially appealing since they are able to represent solutions in the two dimensional plane, since humans are adept at interpreting planar diagrams. Along with PCA, there are some sophisticated dimension-reduction methods such as self-organizing maps (SOM) [9], and Sammon Mapping [14] can map high dimensional data points onto two dimensional plane. However, these methods do not keep account of the mutually non-dominating nature of solutions lying on the PF. Therefore, in general, do not preserve the dominance relations between individual solutions.

The existing approaches do not take into account the practical aspects of a PF solutions for decision making. For example, one might want to know what are the extreme solutions that optimizes only one objective. Another instance being which solution offer most trade-off? If there are multiple clusters in the PF then what solutions compose the boundaries in those cluster? And how are we going to visualize them? These questions are not generally addressed in the current multivariate data visualization techniques.

In this paper, we are going to present an alternative idea to visualize the Pareto front in high dimension by keeping those questions in mind. Moreover, in order to address the visualization of the different aspect of the PF, we will provide some benchmark objective functions that focus on those properties.

2 EXISTING MOOP VISUALIZATION TECHNIQUES

Despite the fact that there are a number of multivariate data visualization techniques, we do not find many examples from the evolutionary multi-objective optimization (EMO) point of view.

Nevertheless, there have been some recent research reports that address this issue from EMO perspective. For example, in [15], the authors present an idea called "Prosection Method". The technique is to project the whole set of solutions to the orthogonal plane where only the solutions from the chosen section are projected. Because multiple planes can be selected for the projection (as in the scatter plot matrix), a prosection matrix is used to visualize all the orthogonal prosections simultaneously. In addition, color coding can be used for distinguishing between feasible and infeasible solutions. However, as this approach depends on the construction of orthogonal plane from multiple objectives to be used as an axis in the final plot, it still suffers from similar limitations as other methods, specially when the dimension is more than four.

We can find other examples in [16] where the authors present a multiple ways to represent a high-dimensional PF. The paper addresses a common problem with the well-known Heatmap visualization. Since an arbitrary ordering of rows and columns renders the Heatmap unclear, the method uses spectral seriation to rearrange the solutions (along with the objective values) and thus enhance the clarity of the Heatmap. A multi-objective evolutionary optimizer is used to further enhance the simultaneous visualization of solutions in objective and parameter space.

In another study [5], the authors proposed a variant of RadViz method that maps data points from a high-dimensional objective space into a 2-D polar coordinate plot while preserving Pareto dominance relationship, retaining shape and location of the PF, and maintaining distribution of individuals. The convergence of the approximate front is measured by radial values of all population members on that front. Meanwhile, the diversity performance is mainly determined by niche count of each subregion in a high-dimensional objective space.

All the existing methods for MOOP solution visualization mainly focus on the visualization of convergence of the PF. Although visualization of the movement of solutions in the search space might be useful for an algorithm designer, however DMs are generally not interested in it. Reducing the number of solution choices and the position of infeasible/robust solutions (with respect to others) are more important to a DM. Therefore, reduction of the cardinality of Pareto solution choices and its intuitive visualization is the main goal of our study.

3 VISUALIZATION FROM A DM'S PERSPECTIVE

Our goal of the visualization method differs from the existing techniques in an interesting way: we are not specifically interested in how close the visualization preserves the topology of the solutions in the actual objective space. As such avenue has already been perused, we argue that such representation is not a principal requirement for decision making. We are also not interested in how the solution converges to the true PF, or visualization of the search trajectory.

Instead, we frame the whole process from a more practical point of view. From our previous experiences with industry collaborations, we have seen that decision making procedure requires a completely different set of criteria than an EMO developer might pursue:

3.1 Solutions with Better Trade-off

First, we need to ask that for a given set of Pareto-optimal solutions (in dimension over 4), what aspects of the solution set is more relevant to look at? For example, the extreme solutions that optimize (i.e. minimize/maximize) a particular objective are not very interesting to look at, therefore if a visualization method can explicitly identify them they will be easier to ignore. A DM will be more interested in the solutions that offer more trade-offs among the objectives. For this reason, the visualization technique should present the interesting (and non-interesting) part of the Pareto-front in an intelligible way. Such that a DM can explicitly choose the potential solution from the presented solution set.

Perhaps, the most desirable aspect of the Pareto-front solutions are those that offer most trade-off among all the points in a solution set. In literature they are known as *Knee points* [2]. If we assume that we do not have any knowledge about the user‘s preferences, it can be justified that the region around that knee is most likely to be interesting for the DM. Since the knee solutions are characterized by the fact that a small improvement in either objective will cause a large loss in the other objective, which makes such movement in either direction not very useful.

Formally, the *knee* solutions are those, from which, moving towards any of the objective axes will cause comparatively higher amount of deterioration in one or more other objectives. In this paper, we will follow the definition of knee points discussed in [1]. Let us consider the simple Pareto-optimal front depicted in Figure 2, with two objectives to be minimized. This front has a clearly visible bulge in the middle. If we assume linear preference functions, and (due to the lack of any other information) furthermore assume that each preference function is equally likely, the solutions at the knee are most likely to be the most desirable choice of the DM. Note that in Figure 2, due to the concavity at the edges, similar reasoning holds for the extreme solutions (edges), which is why these should be considered knees as well. The goal of this paper is to how we should represent such knee solutions if the Pareto-front is constructed in a high dimensional space.

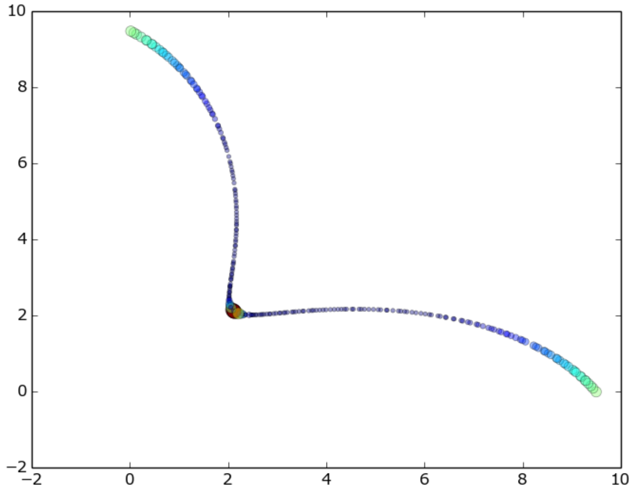


Figure 2: A simple Pareto-optimal front with a knee [1]. The knee and the two extreme solutions are presented by circles with increasing radius.

3.1.1 Pareto-front Generator Function for Knee Identification (DEBMD1K):. The first problem DEBMD1K is a slightly modified problem described in [1]. The problem is scaled version of DEBM1DK, where it can compute Pareto surface for M dimensions and the total number of independent variables will be $N = M - 1$. The function is defined in the equation 1

$$\min f_1(\mathbf{x}) = g(\mathbf{x})r(\mathbf{x})\sin\left(\frac{\pi x_1}{2}\right)\sin\left(\frac{\pi x_2}{2}\right)\ldots\sin\left(\frac{\pi x_{M-1}}{2}\right) \tag{1}$$

$$\min f_2(\mathbf{x}) = g(\mathbf{x})r(\mathbf{x})\sin\left(\frac{\pi x_1}{2}\right)\sin\left(\frac{\pi x_2}{2}\right)\ldots\cos\left(\frac{\pi x_{M-1}}{2}\right)$$

$$\vdots$$

$$\min f_M(\mathbf{x}) = g(\mathbf{x})r(\mathbf{x})\cos\left(\frac{\pi x_1}{2}\right)$$

where,

$$
\begin{aligned}
g(\mathbf{x}) &= 1 + \frac{9}{N-1}\sum_{i=2}^{N} x_i, \quad r(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} r_i(x_i, K) \\
r_i(x_i, K) &= 5 + 10\left(x_i - \frac{1}{2}\right)^2 + \frac{2}{K}\cos(2K\pi x_i) \\
K &= \#\text{ of knees} \\
0 \le x_i &\le 1 \quad i = 1, 2, \ldots, N
\end{aligned}
$$

According to the equation, the knee point is located at $\mathbf{x} = [0.5, 0.5,$
$\ldots, 0.5]$. This fixed location of the knee is useful to verify how the neighborhood embedding distorts the underlying topology in higher dimension.

3.2 Boundary Solutions:

A DM might also be interested to see what are the boundary solutions in a PF, because there is no solution exists beyond the boundary. The boundary solutions are important things to know if the PF is consisted of multiple isolated clusters and the boundary solutions reside on the boundary of a cluster. Hence they are helpful to identify robust solutions in the PF.

3.2.1 Pareto-front Generator Function for Boundary Identification (DTLZ64P):. The DTLZ64P problem is a slightly modified version of DTLZ6 [3]. It is formulated in such a way that in any M dimensions, there will always be 4 disconnected patches.

$$
\begin{aligned}
\min f_1(\mathbf{x}) &= g(\mathbf{x})x_1 \qquad (2)\\
\min f_2(\mathbf{x}) &= g(\mathbf{x})x_2 \\
&\vdots \\
\min f_M(\mathbf{x}) &= g(\mathbf{x})(M - h(\mathbf{x})) \\
&\text{where,} \\
h(\mathbf{x}) &= \sum_{i=1}^{M-1} x_i(1 + \sin(k\pi x_i)), \quad g(\mathbf{x}) = 1 + \frac{9}{N-1}\sum_{i=M}^{N} x_i \\
k &= \begin{cases} 3, & \text{if } i = 1 \\ 1.5, & \text{otherwise} \end{cases} \\
0 \le x_i &\le 1 \qquad i = 1, 2, \ldots, N
\end{aligned}
$$

3.3 Outlier Solutions (OUTLIERPF)

The next thing we wanted to see if the Pareto front has outliers solution and how we can represent them to the DM. In many engineering optimization problems, the search space might be divided into two such region so that one region contains a very small number of solutions and the other with a denser distribution of many points. Moreover, these two regions might be separated from each other very far. The challenge of an evolutionary multi-objective optimization algorithm is to find solutions from both of the regions.

3.3.1 Pareto-front Generator Function for Outlier Identification (OUTLIERPF):. To construct such Pareto front, we formulate a special function for this. We call this function

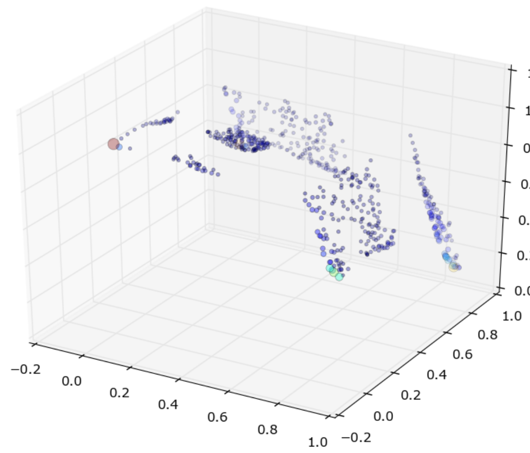


Figure 3: An example three dimensional Pareto-front for the problem OUTLIERPF. The outlier solutions are located near $f_i = 0.0$, where $i \neq M$.

OUTLIERPF:

$$
\begin{aligned}
\min f_1(\mathbf{x}) &= g(\mathbf{x})x_1 \qquad (3)\\
\min f_2(\mathbf{x}) &= g(\mathbf{x})x_2 \\
&\vdots \\
\min f_M(\mathbf{x}) &= g(\mathbf{x})h(\mathbf{x}) \\
&\text{where,} \\
h(\mathbf{x}) &= \frac{1}{100(M-1)}\sum_{i=1}^{M-1}(100 + e^{5x_i}\sin(k\pi x_i)) \\
g(\mathbf{x}) &= 1 + \frac{9}{N-1}\sum_{i=M}^{N} x_i \\
k &= \begin{cases} 4, & \text{if } i = 1 \\ 2, & \text{otherwise} \end{cases} \\
0 \le x_i &\le 1 \quad i = 1, 2, \ldots, N
\end{aligned}
$$

For the three dimensional case (i.e. $M = 3$), the problem OUTLIERPF is composed of five patches of data point clouds. Four of them are closely places and the fifth one is located at $f_i = 0.0$, where $i \neq M$ which is basically the "outliers" in our case. The problem is defined in such a way that irrespective of the dimensions, there will always be such five patches. So that we can verify if the visualization technique can correctly represent this aspect of the Pareto front. An example Pareto-front for this problem for $M = 3$ is presented in the Figure 3.

All these properties can easily be rendered and understood in three dimensional space (i.e. 3 objective problems). However, when the number of objectives become increased, the identification and representation of solutions for knee, boundaries and outliers becomes challenging. In the next section, we discuss our visualization method.

4 VISUALIZATION METHOD

Our method relies on the mapping of high dimensional PF objective values onto lower dimensional space (preferable on a two-dimensional space). During this mapping, we need to ensure that the neighborhood relations among the points be kept as close the points in the original space as possible. If we have a mapping that preserve most of the the local structure of the data points from the higher dimensional space, properties like knee, boundaries and outliers can be visualized in an intelligible way.

4.1 Stochastic Neighborhood Embedding (SNE)

We utilize a method called *Stochastic Neighborhood Embedding (SNE)* [10] [6]. SNE is a probabilistic approach that can place data points, described by high-dimensional vectors or by pairwise dissimilarities, in a low-dimensional space in a way that preserves neighborhood relations. A Gaussian is centered on each solution in the high-dimensional space and the densities under this Gaussian (or the given dissimilarities) are used to define a probability distribution over all the potential neighbors of the solution. The aim of the embedding is to approximate this distribution as well as possible when the same operation is performed on the low-dimensional "images" of the data points. A natural cost function is a sum of Kullback-Leibler divergences, one per solution, which leads to a simple gradient for adjusting the positions of the low-dimensional images. Unlike other dimensionality reduction methods, this probabilistic framework makes it easy to represent each point by a mixture of widely separated low-dimensional images.

Although SNE can construct reasonably good visualizations, it is limited by a cost function that is difficult to optimize and also it suffers from the so called *crowding problem*. In this paper, we have used a variant of SNE called *t-Distributed SNE (t-SNE)* that aims to alleviate these problems. The cost function used by t-SNE differs from the one used by SNE in two ways: (1) it uses a symmetricized version of the SNE cost function with simpler gradients, and (2) it uses a Student-t distribution rather than a Gaussian to compute the similarity between two points in the low-dimensional space. t-SNE employs a heavy-tailed distribution in the low-dimensional space to alleviate both the crowding problem and the optimization problems of SNE. t-SNE is capable of capturing much of the local structure of the high-dimensional data very well, while also revealing global structure such as the presence of clusters at several scales.

4.2 Identification of Knee Solutions in the Higher Dimensional Objective Space

For the identification of the knee solutions, we will follow the method presented in [12]. The definition of Pareto-optimality is a non-reflexive, transitive, and antisymmetric binary relation, i.e., Pareto dominance. Pareto dominance is defined in the following manner:

Given a set of objective functions $\mathbf{F} = [f_1, f_2, \ldots, f_M]$ to be minimized, the vector $\mathbf{F}(\mathbf{x}_i)$ is said to dominate another vector $\mathbf{F}(\mathbf{x}_j)$, denoted $\mathbf{F}(\mathbf{x}_i) \prec \mathbf{F}(\mathbf{x}_j)$, if and only if $f_k(\mathbf{x}_i) \leq f_k(\mathbf{x}_j)$ for all $k \in \{1, 2, \ldots, M\}$ and $f_m(\mathbf{x}_i) < f_m(\mathbf{x}_j)$ for some $m \in \{1, 2, \ldots, M\}$. A point $\mathbf{x}^* \in S$ is said to be *globally Pareto optimal* or a *globally efficient* for a multi-objective optimization problem (MOP) if and only if there does not exist $\mathbf{x} \in S$ satisfying $\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{x}^*)$. $\mathbf{F}(\mathbf{x}^*)$ is then called *globally non-dominated* solution.

Trade-off can be computed over a pair of non-dominated objective vectors and may be defined as the net gain of improvement in some objectives offset by the accompanying deterioration in other objectives as a result of substituting one objective vector with another non-dominated objective vector. Mathematical definition of trade-off is commonly given for every pair of objective functions. The equation to compute this quantity has been borrowed from [12]:

$$T(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{1 \leq m \leq M} \max[0, f_m(\mathbf{x}_j) - f_m(\mathbf{x}_i)]}{\sum_{1 \leq m \leq M} \max[0, f_m(\mathbf{x}_i) - f_m(\mathbf{x}_j)]} \qquad (4)$$

In the definition of $T(\mathbf{x}_i, \mathbf{x}_j)$, the numerator evaluates the total improvement gained by exchanging $\mathbf{x}_j$ with $\mathbf{x}_i$ while the denominator evaluates the deterioration caused by the exchange. The actual metric to evaluate the worth of a solution $\mathbf{x}_i \in R \subset S$ (where, R is an ϵ neighborhood around $\mathbf{x}_i$) in terms of performance trade-off, is given in the following equation:

$$\mu(\mathbf{x}_i, R) = \min_{\forall j: \mathbf{x}_j \in R, \mathbf{x}_i \nprec \mathbf{x}_j, \mathbf{x}_j \nprec \mathbf{x}_i} T(\mathbf{x}_i, \mathbf{x}_j) \qquad (5)$$

A solution with a larger value of the quantity $\mu(\mathbf{x}_i, R)$, within a neighborhood of R, signifies if the solution belong to the knee region. If solution far from the knee region the μ value will be smaller.

Algorithm 1 Extract $\mathcal{PF}$ Boundaries

```
Require: $\mathcal{C} \leftarrow \{C_1, C_2, \ldots, C_m\}$ clusters found from applying DBSCAN on the PF solutions.
Require: $K \leftarrow$ number of boundary solutions required by the DM for each cluster.
$\mathcal{B} \leftarrow \emptyset$, set of boundary points
for each $C_i \in \mathcal{C}$ do
    Find eigenvectors ($\mathbf{v}$) and eigenvalues ($\lambda$) of $C_i$
    $E \leftarrow \{(\mathbf{v}_1, \lambda_1), (\mathbf{v}_2, \lambda_2), \ldots, (\mathbf{v}_n, \lambda_n)\}$
    $E \leftarrow$ sort $E$ according to descending $\lambda$
    for $j$ from 1 to $n$ do
        $V \leftarrow \{\mathbf{v}_j, -\mathbf{v}_j, \mathbf{u}_j, -\mathbf{u}_j\}$    s.t. $\mathbf{u}_j \cdot \mathbf{v}_j = 0$
        for each $\mathbf{w} \in V$ do
            $\mathbf{x} \leftarrow \min_{\mathbf{y} \in C_i} \frac{\mathbf{y} \cdot \mathbf{w}}{\|\mathbf{y}\| \|\mathbf{w}\|}$
            if $|\mathcal{B}| < K$ then
                $\mathcal{B} \leftarrow \mathcal{B} \cup \mathbf{x}$
            else
                return $\mathcal{B}$
            end if
        end for
    end for
end for
```

4.2.1 *Caveat in the Existing Technique:*

Although the measurement made by the equation 5 is quite accurate tool to decide if a solution belongs to the knee region or not, what we have found is that if the number of M becomes larger, the computation of μ becomes more difficult for these reasons:

- ϵ neighborhood is a parameter for μ, which is difficult to infer, since this value depends on the topology and structure of the Pareto-front.
- If the number of objectives is bigger and the total number of solutions in the Pareto-front is smaller, μ function becomes more unreliable. Because of the sparsity of points in the neighborhood.
- Also, we have found the neighborhood radius, i.e. ϵ needs to be different in the different part of the Pareto-front to correctly compute the μ function.

In order to identify and visualize the knee solutions (along with other aspects of the Pareto-front like boundaries etc.), first we normalize all the objective vectors to $[0.0, 1.0]$ range, then we map the normalized high-dimensional Pareto-front onto a low dimensional (in our case, it's 2D) space using t-SNE. After the mapping is done, we use the t-SNE neighborhood information to compute the μ function for each data point. Using this approach, the neighborhood radius can be kept fairly constant, for all our empirical experiments, we have seen this neighborhood radius to be of $0.15 \sim 0.25$. Moreover, this mapping also allows to see the whole Pareto-front in a nicely laid out scatter plot. The results of this visualizations are presented in the Figure 4.

4.3 Identification of Boundaries

In this case, the idea is to find the furthest points on the cluster from the centroid along a number of directions rotated at a specific interval starting from the principal axis. Let's say we want to get K boundary points on a cluster. First, for a given cluster, we find it's eigenvectors (and their corresponding eigenvalues). Then we draw a line from the cluster centroid along the direction of the eigenvector (with the highest eigenvalue) and find the points on the cluster that is the furthest from the centroid that is on the line or having smallest normal distance from the line. Then we rotate the line by a certain angle and do the same. In our case, we have taken four orthogonal lines, i.e. four boundary points orthogonal to each other. Then we pick the eigenvector with the next biggest eigenvalue and repeat the same process. We keep doing it until we have extracted K boundary points and show them on the t-SNE scatter plot at the end. The value of K can be specified by a DM. The overall boundary extraction algorithm is presented in Algorithm 1. An example boundary extraction procedure is presented in the Figure 5, where the PF is generated from the engineering design problem described in [11], which we call as CAR-CRASH problem. Its PF is consisted of 3 clusters and the boundary extraction algorithm operates on each of the clusters.

4.4 Identification of Outliers

For detection of outliers. We need to find the isolated patches of point clouds in the higher dimensional space. Therefore, a straight-forward approach should be to apply clustering algorithm. However, if the number of objective is high the clustering algorithm might not identify the outliers in a correct way. To alleviate this problem, we can apply the clustering method after the t-SNE mapping. However, we have found that t-SNE does not always correctly map the points to preserve the global topological relation onto low dimensional space.

The t-SNE algorithm adapts its realization of "distance" to local density variations in the data set. As a result, it naturally expands dense clusters, and contracts sparse ones, blurring out cluster sizes. As a result, the t-SNE mapping distorts the actual cluster sizes of the original data. Also, the global geometry of the original data points depends on the parameter called *perplexity*. For different perplexity values, the inter-cluster distances come out to be different. Therefore, mapping produced by t-SNE is not very suitable for cluster analysis. For clustering we have used DBSCAN algorithm [4].

5 VISUALIZATION RESULTS

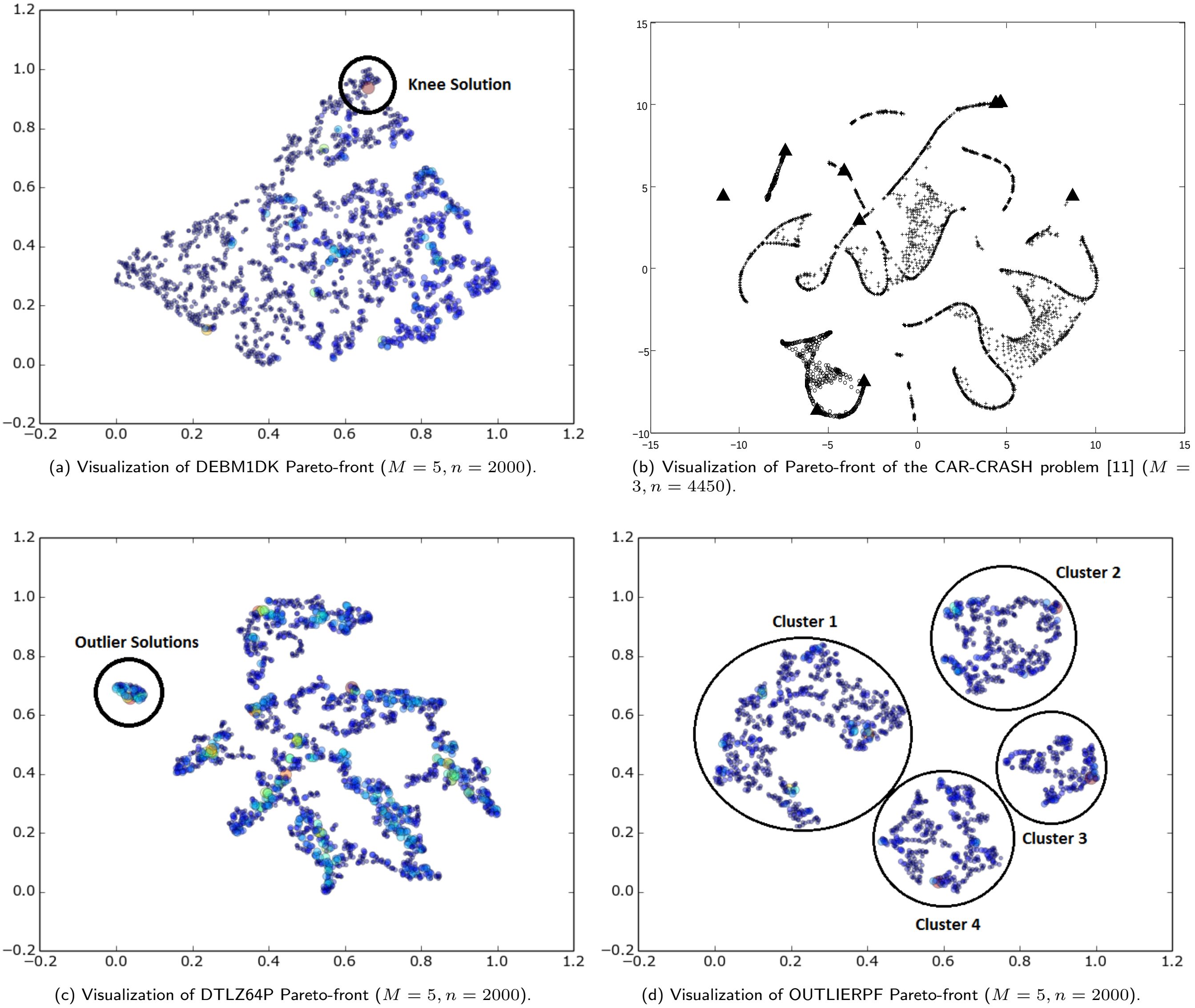
The visualization results are presented in the Figure 4. Here we present the results from the Pareto-front generator functions described in this paper. The plots are presented as scatter plots found from the t-SNE mappings where each point is represented with a circle and the radius of each circle is proportional to corresponding mu value of a solution. A larger radius represents higher mu value, hence they are on the knee of the PF. The Figure 4a presents the visualization of DEBM1DK problem for 5 objective case. The largest circle on the top portion of the scatter plot is the knee point and the other larger circles are mapped from the extreme solutions of the PF.

The visualization for DTLZ4P is presented in the Figure 4c. Here we can see four identifiable clusters (PF patches) in the t-SNE mapping, the knee solutions are also equally identifiable. The Figure 4d presents the visualization for OUTLIERPF problem. The small cluster on the top left corner of the scatter plot is the outliers in the PF. This Pareto-front has multiple spots where the trade-offs are comparatively better than the rest of the solutions.

The boundaries from each of the clusters from the PF of the CAR-CRASH problem [11] is presented in the Figure 4b. The boundary points are presented with large circles circles.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have demonstrated an alternative approach to address the issue of high dimensional Pareto-front visualization. Our approach is more practical in a sense that it takes account of the DM's perspective in the visualization mechanism. Although our method is not suitable for visualizing an EMO convergence, however such convergence plots are



(a) Visualization of DEBM1DK Pareto-front ($M = 5, n = 2000$).

(b) Visualization of Pareto-front of the CAR-CRASH problem [11] ($M = 3, n = 4450$).

(c) Visualization of DTLZ64P Pareto-front ($M = 5, n = 2000$).

(d) Visualization of OUTLIERPF Pareto-front ($M = 5, n = 2000$).

Figure 4: The t-SNE mapping of different high dimensional Pareto front of different problems. Here M is the dimension of the objective space and n is the number of solutions (i.e. objective vector/data points). The radius of each data point is proportional to the μ value computed by the Equation 5. For the CAR-CRASH problem, the cluster are shown in different markers and the filled triangles are the boundary points.

not very useful during the decision making since other metric based visualizations (function evaluation vs. hypervolume or GED) are good enough for such purposes. In spite of some limitations regarding the correct identifications of clusters in the PF, we present our idea as an initial "proof of concept" to tackle the issue of visualization in MOOP. Our approach is mainly based on the t-SNE mapping which does not retain the global topological relations among the data points. In the future we would like to improve the current implementation by adopting/modifying other mapping techniques like Locally Linear Embedding [13].

The method of knee finding technique can also be used to find the interesting regions in the high dimensional PF. For example, first using a small number of population we can sample the PF using a generic many-objective optimization algorithms. From that limited sample, we will be able to locate the regions where the knee solutions are more likely to be found. After that we can use deterministic method/scalarization methods like AASF to find more solutions in the knee region.

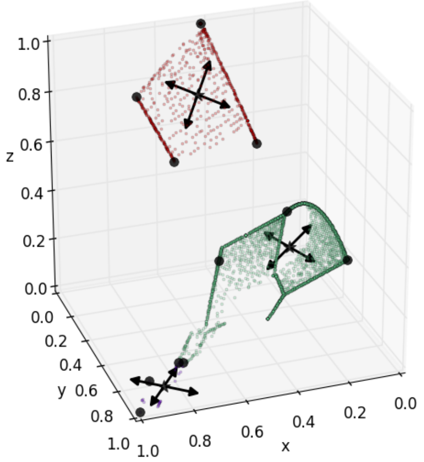


Figure 5: The boundary extraction mechanism procedure on the PF for the engineering design problem taken from [11]. DBSCAN algorithm is able to find three clusters and the boundary points are presented with large circles. The arrows correspond to two principal axes, as a result algorithm picks four boundary points from each cluster.

In this way we will be able to find the solutions on the PF that matters the most to the DM thus help saving a large amount of computational effort to find the entire PF.

REFERENCES

[1] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. 2004. Finding Knees in Multi-objective Optimization. In *Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference, Birmingham, UK, September 18-22, 2004. Proceedings*, Xin Yao, Edmund K. Burke, José A. Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiňo, Ata Kabán, and Hans-Paul Schwefel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 722–731. https://doi.org/10.1007/978-3-540-30217-9_73

[2] Indraneel Das. 1999. On characterizing the "knee" of the Pareto curve based on Normal-Boundary Intersection. *Structural optimization* 18, 2 (1999), 107–115. https://doi.org/10.1007/BF01195985

[3] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. 2002. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, Vol. 1. Institute of Electrical & Electronics Engineers (IEEE), Piscatway: NJ, 825–830. https://doi.org/10.1109/cec.2002.1007032

[4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI Press, Palo Alto: CA, 226–231.

[5] Zhenan He and Gary G. Yen. 2016. Visualization and Performance Metric in Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 20, 3 (jun 2016), 386–402. https://doi.org/10.1109/tevc.2015.2472283

[6] Geoffrey E Hinton and Sam T. Roweis. 2003. Stochastic Neighbor Embedding. In *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer (Eds.). MIT Press, Cambridge, MA, 857–864. http://papers.nips.cc/paper/2276-stochastic-neighbor-embedding.pdf

[7] Patrick Hoffman, Georges Grinstein, and David Pinkney. 1999. Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations. In *Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation in Conjunction with the Eighth ACM Internation Conference on Information and Knowledge Management (NPIVM '99)*. ACM, New York, NY, USA, 9–16. https://doi.org/10.1145/331770.331775

[8] A. Inselberg. 2009. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, Berlin: Germany.

[9] T. Kohonen, M. R. Schroeder, and T. S. Huang (Eds.). 2001. *Self-Organizing Maps* (3rd ed.). Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[10] Geoffrey Hinton Laurens van der Maaten. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (nov 2008), 2579–2605.

[11] Xingtao Liao, Qing Li, Xujing Yang, Weigang Zhang, and Wei Li. 2007. Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Struct Multidisc Optim* 35, 6 (nov 2007), 561–569. https://doi.org/10.1007/s00158-007-0163-x

[12] Lily Rachmawati and Dipti Srinivasan. 2009. Multiobjective Evolutionary Algorithm With Controllable Focus on the Knees of the Pareto Front. *IEEE Transactions on Evolutionary Computation* 13, 4 (Aug 2009), 810–824. https://doi.org/10.1109/tevc.2009.2017515

[13] Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290, 5500 (2000), 2323–2326. http://www.jstor.org/stable/3081722

[14] J. W. Sammon. 1969. A Nonlinear Mapping for Data Structure Analysis. *IEEE Trans. Comput.* C-18, 5 (May 1969), 401–409. https://doi.org/10.1109/T-C.1969.222678

[15] Tea Tusar and Bogdan Filipic. 2015. Visualization of Pareto Front Approximations in Evolutionary Multiobjective Optimization: A Critical Review and the Prosection Method. *IEEE Transactions on Evolutionary Computation* 19, 2 (apr 2015), 225–245. https://doi.org/10.1109/tevc.2014.2313407

[16] David J. Walker, RichardM. Everson, and Jonathan E. Fieldsend. 2013. Visualizing Mutually Nondominating Solution Sets in Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 17, 2 (apr 2013), 165–184. https://doi.org/10.1109/tevc.2012.2225064